Producing Monolingual and Parallel Web Corpora at the Same Time – SpiderLing and Bitextor's Love Affair

Nikola Ljubešić,[*] Miquel Esplà-Gomis,[†] Antonio Toral,[§] Sergio Ortiz-Rojas,[‡] Filip Klubička[*]

[*]Dept. of Information and Communication Sciences,
University of Zagreb, Zagreb (Croatia)
nljubesi@ffzg.hr
fklubicka@ffzg.hr

[†]Dept. de Llenguatges i Sistemes Informàtics,
Universitat d'Alacant, Alacant (Spain)
mespla@dlsi.ua.es

[§]ADAPT Centre, School of Computing,
Dublin City University, Dublin (Ireland)
atoral@computing.dcu.ie

[‡]Prompsit Language Engenering,
Elx (Spain)
sergio@prompsit.com

Abstract

This paper presents an approach for building large monolingual corpora and, at the same time, extracting parallel data by crawling the top-level domain of a given language of interest. For gathering linguistically relevant data from top-level domains we use the SpiderLing crawler, modified to crawl data written in multiple languages. The output of this process is then fed to Bitextor, a tool for harvesting parallel data from a collection of documents. We call the system combining these two tools Spidextor, a blend of the names of its two crucial parts. We evaluate the described approach intrinsically by measuring the accuracy of the extracted bitexts from the Croatian top-level domain `.hr` and the Slovene top-level domain `.si`, and extrinsically on the English–Croatian language pair by comparing an SMT system built from the crawled data with third-party systems. We finally present parallel datasets collected with our approach for the English–Croatian, English–Finnish, English–Serbian and English–Slovene language pairs.

Keywords: crawling, top-level domain, monolingual corpus, parallel corpus

1 Introduction

Parallel data are one of the most important linguistic resources for cross-lingual natural language processing (Melamed, 2001). Parallel corpora consist of collections of texts in different languages which are mutual translations. This resource is specially relevant in the field of statistical machine translation (SMT), where parallel corpora are used to learn translation models automatically. The growing interest in SMT in the last decades has increased the demand of parallel corpora and, as a consequence, new strategies have been proposed to collect such data. Many sources of bitexts have been identified; some examples are:

- texts from multilingual institutions, such as the Hansards corpus (Roukos et al., 1995) or the Europarl corpus (Koehn, 2005);
- translations of software interfaces and documentation, such as KDE4 and OpenOffice (Tiedemann, 2009); or
- news translated into different languages, such as the SETimes corpus (Ljubešić, 2009), or the News Commentaries corpus (Bojar et al., 2013).

However, one of the most obvious sources for collecting parallel data is the Internet. On the one hand, most of the sources already mentioned are currently available on the Web. In addition to this, it is worth noting that many websites are available in several languages and this translated content is another useful source of parallel data. Therefore, a considerable scientific effort has been put during the last years in order to exploit the web as a source to automatically acquire new parallel data (see Section 2). Some examples of corpora built from multilingual web pages are the *Tourism English–Croatian Parallel Corpus 2.0*[1] (Toral et al., 2014) or the *Panacea* project's parallel corpora for English–French and English–Greek in two different domains: environment[2] and labour legislation[3] (Pecina et al., 2014).

There are several tools that can be used for automatically crawling parallel data from multilingual websites (Papavassiliou et al., 2013; Esplà-Gomis and Forcada, 2010). However, all of them share the same limitation: they require the user to provide the URLs of the multilingual websites to be crawled. Despite the fact that large amounts of parallel data can be obtained from a single website, this requirement implies that these tools will require a list of web pages to crawl and will not be able to exploit the web as a parallel corpus in a fully automated way.

To deal with this limitation, we propose a new method that focuses on crawling top-level domains (TLD) for multilingual data, and then detects parallel data inside the crawled data. We implement this method in a tool called Spidextor, a name that is a result of blending the names of the two tools which are the base of this system: SpiderLing (Suchomel et al., 2012), a monolingual crawler that focuses on

[1] `http://hdl.handle.net/11356/1049`

[2] `http://catalog.elra.info/product_info.php?products_id=1182`

[3] `http://catalog.elra.info/product_info.php?products_id=1183`

linguistically-relevant content and is able to crawl a whole TLD, and Bitextor, a parallel data crawler that is able to detect translated documents on crawled websites. The combination of these two tools allows to obtain: (a) a huge amount of multilingual data that is assumed to be linguistically relevant, and (b) as much parallel data as possible from this multilingual data. This process is carried out in a fully automatic fashion. In addition, it is worth mentioning that both monolingual and parallel data are the base for building SMT systems, which makes Spidextor especially interesting for this field.

In this paper, we describe four parallel corpora built with Spidextor for four language pairs: English–Croatian, English–Finnish, English–Serbian, and English–Slovene. In addition, we evaluate the quality of the English–Croatian parallel corpus built with Spiderling by carrying out two different evaluations on it: one intrinsic, by evaluating directly the quality of the corpus built, and one extrinsic, by building new SMT systems from crawled corpora and evaluating their performance, comparing them to third-party systems.

The rest of the paper is organised as follows: Section 2 describes the main approaches to the problem of parallel data crawling. Section 3 describes the tool Spidextor. Section 4 describes the new corpora crated for English–Croatian, English–Slovene, English–Serbian, and English–Finnish language pairs, while Section 5 describes the evaluation carried out for these new resources and the results obtained. The paper ends with some concluding remarks in Section 6.

2 Related work

One of the most usual strategies to crawl parallel data from the Internet is to focus on web sites that make it straightforward to detect parallel documents (Nie et al., 1999; Koehn, 2005; Tiedemann, 2012). Many approaches use content-based metrics (Jiang et al., 2009; Utiyama et al., 2009; Yan et al., 2009; Hong et al., 2010; Sridhar et al., 2011; Antonova and Misyurev, 2011; Barbosa et al., 2012), such as bag-of-words overlapping. Although these metrics have proved to be useful for parallel data detection, their main limitation is that they require some linguistic resources (such as a bilingual lexicon or a basic machine translation system) which may not be available for some language pairs. To avoid this problem, other works use the HTML structure of the web pages, which usually remains stable between different translations of the same document (Ma and Liberman, 1999; Nie et al., 1999; Resnik and Smith, 2003; Zhang et al., 2006; Désilets et al., 2008; Esplà-Gomis and Forcada, 2010; San Vicente and Manterola, 2012; Papavassiliou et al., 2013). Another useful strategy is to identify language markers in the URLs (Ma and Liberman, 1999; Nie et al., 1999; Resnik and Smith, 2003; Zhang et al., 2006; Désilets et al., 2008; Esplà-Gomis and Forcada, 2010; San Vicente and Manterola, 2012) that help detecting possible parallel documents.

Some authors have used similar approaches to crawl comparable corpora. For instance, Smith et al. (2010) use the links between translated articles in Wikipedia to crawl parallel sentences or words. A more complex strategy is used by Munteanu and Marcu (2005), who compare news published in versions of news websites written in different languages by using a publication time stamp window. In this way, it is possible to retrieve likely on-topic news on which a cross-lingual information retrieval strategy is applied based on word-to-word machine translation.

Even though these methods have proven to be useful for specific web sites, the real challenge is to find strategies that allow to extend them to crawl the Web in an unsupervised fashion, therefore allowing to exploit the real potential of this resource. Resnik (1998) uses language anchors, i.e. phrases in a given language that may be a hint indicating that the translation of a web page is available through a link, such as the link captions “in Chinese” or “Chinese” in a website in English. Resnik (1998) builds queries containing two possible anchors in two languages and queries the *Altavista* search engine to find potentially parallel websites. Chen and Nie (2000) use a similar approach, but they look for anchors separately, i.e. in two different queries for each language. Once this is done, the URLs in both results are compared in order to obtain the lists of websites that might contain parallel documents. Ma and Liberman (1999) use a more direct approach; they download the list of websites of a given top level domain (TLD), download each of them, and apply language identification to keep only the documents in the languages desired. Similarly, Resnik and Smith (2003) use the Internet Archive[4] to obtain a list of URLs for several specific TLDs. A set of rules are then applied on the URLs of the different TLDs in order to find parallelisms between them and, therefore, candidate parallel documents. Smith et al. (2013) extend this approach to use it on the Common Crawl corpus (Spiegler, 2013).

In this paper we propose a novel strategy for building both parallel and monolingual corpora automatically by crawling TLDs. This strategy consists in combining two different existing tools: the SpiderLing monolingual crawler (Suchomel et al., 2012), which is able to automatically harvest documents from a given TLD starting from a collection of seed URLs, and the Bitextor parallel data crawler (Esplà-Gomis et al., 2014). The main differences between this approach and other previous works are as follows: (i) this method is aimed at crawling both monolingual and parallel data in the same process, an objective that is especially convenient for some natural language processing problems such as SMT, and (ii) this approach allows to obtain parallel data in a totally automatic fashion, i.e. without having to provide the specific URLs that are likely to contain the parallel data to be crawled.

3 Spidextor

In this section we present Spidextor (a blend of the names of its two crucial parts – SpiderLing and Bitextor), the tool that we have developed that enables us to crawl a TLD for documents written in specific languages, and subsequently match documents written in different languages that are probably translations of each other. Figure 1 shows the structure of the process carried out to obtain the new corpora described in Section 4.

[4] `https://archive.org`

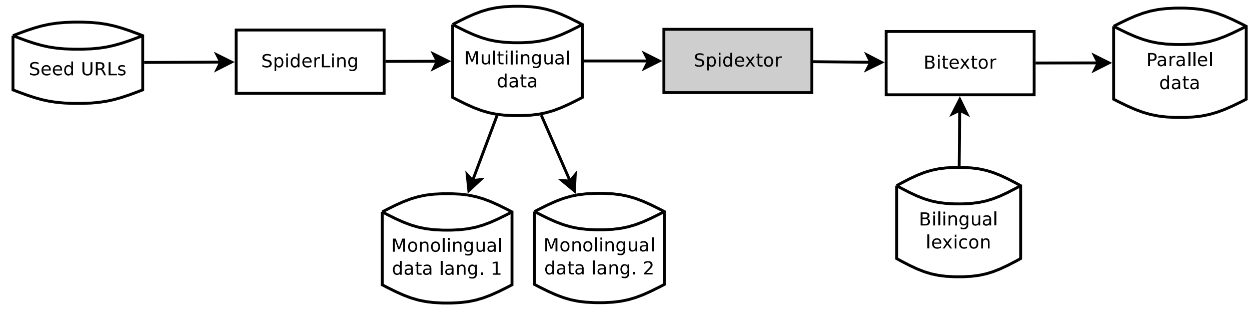

Figure 1: Structure of the process carried out by the combination of the tools SpiderLing and Bitextor with Spidextor.

3.1 SpiderLing modifications

For crawling the TLD we use the SpiderLing crawler version 0.77,[5] which is part of the `Brno` pipeline for producing big, clean and cheap web corpora (Suchomel et al., 2012). SpiderLing was primarily built to produce monolingual corpora. Minor modifications of the code (20 lines added or modified) had to be introduced to enable the user to define multiple languages of interest. Thereby, all documents written in any of the languages are kept in the crawl.

We do not distribute our alternations in the code as they were adopted in the official SpiderLing version 0.82.[6]

Since SpiderLing uses a simple distance-based language identification procedure (as it was meant to discriminate between documents written in the language of interest and all other languages), having now multiple languages in our crawl, we included in our process one additional language identification with `langid.py`[7] on the output of SpiderLing to double check the predicted language and filter out or reclassify the wrong predictions.

3.2 Bitextor integration

Bitextor is oriented to process single websites, therefore some adaptations were necessary to process the number of websites containing documents in the target languages collected while crawling a top-level domain. Given the size of the data collected by SpiderLing, the adaptation has been done with multiprocessing in mind, and the resulting procedure was able to process the Finnish TLD crawl (18 million html documents) on a machine with 64GB of RAM and 16 cores in just 5 hours.

The logic added on top of Bitextor necessary to process the SpiderLing output consists of two scripts only: a script that transforms the SpliderLing output (`.prevert_d`[8] files) to Bitextor's `.lett` format[9] and another configuration script that enables the user to define the language pairs he/she is interested in, together with all the paths required to run Bitextor, one of which is a small bilingual lexicon which can improve the bitext extraction results.[10] The first script also produces and runs a Makefile in a parallel fashion. All Bitextor's processing is run on on the level of each Internet domain, making the parallelisation of the process straightforward.

The output of the Bitextor processing are `.tmx` and `.txt` files consisting of the parallel candidates, organised by domains.

4 Resulting resources

The four TLDs on which we focused in this work (`.fi`, `.hr`, `.sr`, and `.si`) were crawled for periods of different length, depending on the size of the domains. While the Slovene domain was crawled for three days only, we crawled the Finnish domain for seven days.

We ran Bitextor on each multilingual domain separately, limiting thereby the search space for parallel data on specific domains. Naturally, parallel data could be found between domains as well, but (1) this is not a frequent case and (2) this limitation of the search space makes the bitext extraction process computationally much less expensive.

The sizes of the resulting parallel corpora are shown in Table 1. The figures in this table correspond to the amount of unique segment pairs and the total number of words contained in both of them. We call the resulting data sets *fienWaC*, *hrenWaC*, *srenWaC*, and *slenWaC* following the corpus naming convention of the *WaCKy* initiative (Baroni et al., 2009).

The corpora obtained are distributed under the *CLARIN.SI END-USER LICENCE FOR INTERNET CORPORA*[11]. These corpora consist of a collection of translation memories in TMX format for the following language pairs:

- English–Croatian (Ljubešić et al., 2016a),[12]

[5] `http://nlp.fi.muni.cz/trac/SpiderLing/attachment/wiki/WikiStart/SpiderLing-src-0.77.tar.xz`

[6] `http://corpus.tools/raw-attachment/wiki/Downloads/spiderling-src-0.82.tar.xz`

[7] `https://github.com/saffsd/langid.py`

[8] `.prevert_d` files contain all the text extracted from the crawling, labelled with information about the original documents from which they were extracted, and the language in which each of them is written.

[9] `.lett` files contain plain text consisting of a line for every document processed. Each line consists of 6 tab-separated values: a two-character language identification, the mime type, the character encoding, the original URL, the HTML content of the document encoded in base64, and the clean plain text in the HTML document.

[10] For all our language pairs we use small bilingual lexicons extracted automatically from phrase tables built on existing parallel data. If no parallel data is available, a simple Internet-like lexicon can be used instead.

[11] `http://www.clarin.si/info/wp-content/uploads/2016/01/CLARIN.SI-WAC-2016-01.pdf`

[12] `http://hdl.handle.net/11356/1058`

corpus	web domains	segments	words
fienWaC	10,664	2,866,574	77,048,083
hrenWaC	5,624	1,554,912	55,083,246
slenWaC	3,529	718,315	27,924,210
srenWaC	2,546	534,682	23,139,804

Table 1: Total number of web domains crawled, number of unique pairs of segments and number of words obtained with Spidextor for the `.hr`, `.fi`, `.si`, and `.sr` TLDs.

corpus	segments	words
fienWaC	4,079,704	100,104,805
hrenWaC	2,444,478	71,724,438
slenWaC	974,334	37,616,705
srenWaC	623,955	27,056,129

Table 2: Total number of segments in the collection of translation memories built for each language pair.

- English–Finnish (Ljubešić et al., 2016b),[13]
- English–Slovene (Ljubešić et al., 2016d),[14] and
- English–Serbian (Ljubešić et al., 2016c).[15]

The total size of translation units for the whole collection of translation memories is shown in Table 2. The difference in the amounts of segments and words between tables 1 and 2 is due to the fact that for the distributed corpora duplicate segment pairs are allowed, given that they may come from different web pages.

5 Resource evaluation

In order to properly evaluate our method for building parallel resources, we performed two flavours of evaluation: one intrinsic and the other extrinsic.[16] Intrinsic criteria are those connected to the goal of the system, i.e. criteria for evaluating the resources directly, whereas the extrinsic ones are connected to the system's function. Thus, in order to do an intrinsic evaluation, it should suffice to manually evaluate the accuracy of a random sample of the corpora obtained. We performed an intrinsic evaluation on the English–Croatian and English–Slovene datasets.

Meanwhile, extrinsic evaluation analyses the system's performance in a broader context of application; in our case, we used our new resources as the training corpus for a SMT system and evaluated it using automatic evaluation metrics. We performed extrinsic evaluation on the English-Croatian pair only.

5.1 Intrinsic evaluation

We performed our intrinsic evaluation on the hrenWaC and the slenWaC corpora by evaluating 100 potential parallel segments per corpus, towards a total of 200 segment pairs.

corpus	granularity	match	partial	miss
hrenWaC	segments	76.0%	4.0%	20.0%
	words	77.7%	6.4%	15.9%
slenWaC	segments	63.0%	4.0%	33.0%
	words	60.8%	7.3%	31.9%

Table 3: Fraction of full matches (match), partial matches (partial) and misaligned segment pairs (miss) in the hrenWaC and the slenWaC corpora, both at the level of words and at the level of segment pairs.

As regards the The hrenWaC corpus, it is based on a crawl of 6.1 million documents acquired from 25,924 domains, from which only 6,228 contained documents both in English and Croatian. From the collection of documents obtained, 10.5% of them were in English, while the rest were in Croatian. Potential parallel data were found by using Bitextor on 5,624 domains. In the case of the enslWaC corpus, it is built from 3.6 million documents crawled from 4,049 domains. Among all the crawled documents, 9.88% of them were written in English, similar as on the Croatian TLD. Parallel data was extracted from 3,529 domains.

The portion of the corpora evaluated were manually inspected to check whether or not they were indeed parallel, or rather, whether one was a translation of the other. To do this, we used a simple annotation schema consisting of three categories: match, partial match and miss. The partial match category was introduced to cover cases where more than 50% but less than 90% of the text in a given pair of segments was parallel.

The results of this evaluation are reported in Table 3. They show that noise is quite present in both resources: ∼16% of words in the English–Croatian resource and ∼32% in the English–Slovene resource. At first the fact that the English–Slovene corpus contains, on word level, double the amount of noise in comparison to the English–Croatian resource can be surprising as (1) the two languages are typologically very close and (2) the percentage of English data in both TLDs is very similar. A probable explanation for the obtained difference in noise levels can lie in the bilingual lexicon used by Bitextor for document alignment. While the Croatian–English lexicon was extracted from quite diverse parallel datasets (hrenWaC v1.0, SETimes, TED talks), the Slovene–English lexicon was extracted from parallel data of a much narrower domain (Europarl, JRC-Acquis).

5.2 Extrinsic evaluation

We perform extrinsic evaluation of the hrenWaC parallel corpus in the scenario in which this dataset is used as a training corpus for building a SMT system. We built new SMT systems using the corpora collected, and compared them to some of the most popular MT systems available on the Internet providing translation between English and Croatian: Yandex.Translate,[17] Bing Translator,[18] and Google Translate.[19] This section describes the details of

[13] `http://hdl.handle.net/11356/1060`

[14] `http://hdl.handle.net/11356/1061`

[15] `http://hdl.handle.net/11356/1059`

[16] This is a well-known approach in evaluating natural language processing tools (Mollá and Hutchinson, 2003; Schneider et al., 2010).

[17] `https://translate.yandex.com`

[18] `http://www.bing.com/translator/`

[19] `https://translate.google.com/`

the evaluation setting defined for the extrinsic evaluation process.

Parallel data. The newly created hrenWaC corpus is evaluated in two different ways in this section:

- building an SMT system trained solely on the hrenWaC corpus, in order to assess the performance that can be obtained using a corpus obtained fully automatically with Spidextor, and

- building an SMT system by combining the hrenWaC corpus with all the freely available English–Croatian parallel corpora, in order to assess the performance that can be obtained when adding new data crawled from a TLD to the already available parallel corpora.

The freely available parallel resources for the English–Croatian language pair at the moment of running our experiments were the following: the DGT-TM (Steinberger et al., 2015) parallel corpus, the JRC-Acquis (Steinberger et al., 2014) parallel corpus, the OpenSubtitles (Tiedemann, 2013) parallel corpus, the SETimes (Ljubešić, 2009) parallel corpus, and the TED talks (M. Cettolo, 2015) parallel corpus. Combining all these corpora with the hrenWaC leads to 19.5 million segments, 16.9 million of them coming from the OpenSubtitles corpus.
When training the SMT system that combines all the available parallel data, we interpolate the translation models built on each parallel dataset via our development data, therefore assuring that the OpenSubtitles parallel corpus, which is both large and noisy, does not interfere with the quality of the final translator. Although the OpenSubtitles corpus does contribute most of the data, in the remaining datasets there is still more than 2 times the amount of data than in the hrenWaC dataset.

Development sets. The development set used in our experiments was created by translating into Croatian a subset (the first 25 news stories, accounting for $1,011$ sentences) of the English side of the test set provided for the Workshop on Statistical Machine Translation in 2012 (WMT12).[20] We obtain translations of this data set in two ways: professional translation and crowdsourcing. While professional translations lead to a higher quality parallel data set, which should result in a positive impact on the final MT output, its cost can be close to an order of magnitude higher than crowdsourcing. All in all we have three translation references in Croatian in the development set; two obtained by using crowdsourcing, and an additional one obtained by means of professional translation. Further details about the way in which these development sets were generated are available in the public deliverable D3.1c of the AbuMaTran project.[21] In the experiments below we use the three references for the direction English→Croatian while only the professional translation is used for the opposite direction. These are the references that led to the best results in the development phase.

Test set. The test set used for evaluating the SMT systems described in this work was based on the test set used in the evaluation campaign of WMT13.[22] This corpus consists of a collection of news stories in English which are freely available, translated into other languages. Unfortunately, Croatian was not one of these languages. Therefore, in order to build a suitable test set, the first $1,000$ segments were manually translated into Croatian by a native speaker.

Models. The SMT systems are phrase-based and built with the Moses toolkit (Koehn et al., 2007) using default parameters, except for the following. On top of the default word-based reordering model, our system implements two additional ones, phrase-based and hierarchical (Galley and Manning, 2008). On top of this, two additional resources were obtained from the hrenWaC parallel corpus and used in the MT systems built: an operation sequence model (Durrani et al., 2011) and a bilingual neural language model (Devlin et al., 2014).

Results. We evaluate the systems trained in both directions and on both data collections (only hrenWaC and all the training corpora) with two widely used automatic evaluation metrics: BLEU (Papineni et al., 2002) and TER (Snover et al., 2006). The results are shown in Table 4. As it can be seen, in both translation directions the performance of the new SMT systems built on the hrenWaC corpus obtain results that are comparable to those obtained by the third-party systems. In the harder translation direction, English→Croatian, the newly built SMT systems outperform two of the reference systems, Bing and Yandex, while we do not observe a substantial decrease in the quality of the MT system built solely on the hrenWaC parallel corpus compared to the system built on all the training corpora.

direction	**system**	**BLEU**	**TER**
en→hr	Google	0.2673	0.5946
	Bing	0.2281	0.6263
	Yandex	0.2030	0.6801
	hrenWaC	0.2457	0.6198
	all	0.2445	0.6147
hr→en	Google	0.4099	0.4635
	Bing	0.3658	0.5199
	Yandex	0.3463	0.5311
	hrenWaC	0.3499	0.5090
	all	0.3721	0.4878

Table 4: This table reports BLEU and TER for the two SMT systems built on the hrenWaC corpus (hrenWaC and all) and the three third-party on-line MT systems (in grey), Google Translate, Bing Translator, and Yandex.Translate, in both translation directions: English into Croatian (en→hr) and Croatian into English (hr→en).

However, in the opposite direction, Croatian→English, there is a significant difference in the performance achieved by both newly built MT systems: the system using all the

[20] `http://www.statmt.org/wmt12/translation-task.html`

[21] `http://www.abumatran.eu/?page_id=59`

[22] `http://matrix.statmt.org/test_sets/newstest2013.tgz?1367361979`

training data achieves a 2.22 BLEU points increase and a 2.12 TER points decrease when compared to that trained only on the hrenWaC corpus. As regards the third-party MT systems, in this case, the MT system trained on all the data available outperforms once more both Bing and Yandex, while the one trained only on the hrenWaC parallel corpus obtains results very close to those obtained by Yandex, but still lower than Bing and Google.

As can be seen in these results, the SMT systems obtained are not able to outperform all the third-party MT systems used for evaluation. However, it is worth mentioning that, given that the data used for building these models was obtained in a fully automatic fashion by crawling TLDs, the results are quite positive, since they show that it is possible to obtain an MT system comparable to some of the most used online MT systems by only running an automatic crawling process for a few days, with the only explicit input being a TLD to be crawled and a small bilingual English–Croatian lexicon.

6 Concluding remarks

In this paper we have presented a strategy for combining two tools, SpiderLing and Bitextor, in order to automatise the process of crawling TLDs to build both monolingual and parallel corpora in a fully-automatic fashion. The combination of both tools is implemented with two scripts that plug the output of SpiderLing (the tool responsible of crawling monolingual corpora from TLDs) to the input of Bitextor (the tool responsible to detect and align parallel data from a crawled website). These scripts are available under GPLv3 license at `https://github.com/abumatran/spidextor/`.

Using this tool, several large parallel corpora have been obtained from TLDs. These corpora, obtainable from the CLARIN.SI repository, cover the following language pairs: English–Croatian, English–Finnish, English–Serbian, and English–Slovene. The English–Croatian parallel corpus has been evaluated in two different ways: with an intrinsic evaluation, that consisted of manually checking a portion of the parallel corpus, and with an extrinsic evaluation, that consisted of building a phrase-based SMT system and evaluating it with standard quality metrics for translation tasks in both translation directions. The results obtained by means of the intrinsic evaluation have proved that Spidextor is able to obtain reasonably clean parallel corpora, with a success rate in segment-level alignment of about 76% in the best case. The extrinsic evaluation has shown that the SMT systems built on parallel corpora collected with Spidextor in this case can obtain results comparable to those obtained by some of the most popular online MT systems. The evaluation carried out in this work confirms that Spidextor allows obtaining all the data needed for training an SMT system with a performance comparable to other commercial systems in a fully automatic fashion.

7 Acknowledgments

This research is supported by the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement PIAP-GA-2012-324414 (AbuMaTran).

8 Bibliographical References

Antonova, A. and Misyurev, A. (2011). Building a web-based parallel corpus and filtering out machine-translated text. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 136–144, Portland, Oregon. Association for Computational Linguistics.

Barbosa, L., Rangarajan Sridhar, V. K., Yarmohammadi, M., and Bangalore, S. (2012). Harvesting parallel text in multiple languages with limited supervision. In *Proceedings of COLING 2012*, pages 201–214, Mumbai, India.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the 8th Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria.

Chen, J. and Nie, J.-Y. (2000). Parallel web text mining for cross-language IR. In *Proceedings of RIAO*, pages 62–77.

Désilets, A., Farley, B., Stojanovic, M., and Patenaude, G. (2008). WeBiText: Building large heterogeneous translation memories from parallel web content. In *Proceedings of Translating and the Computer*, pages 27–28, London, UK.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J. (2014). Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *Proceedings of ACL*, pages 1370–1380.

Durrani, N., Schmid, H., and Fraser, A. (2011). A Joint Sequence Translation Model with Integrated Reordering. In *Proceedings of ACL/HLT*, pages 1045–1054.

Esplà-Gomis, M. and Forcada, M. L. (2010). Combining content-based and URL-based heuristics to harvest aligned bitexts from multilingual sites with bitextor. *The Prague Bulletin of Mathematical Linguistics*, 93:77–86.

Esplà-Gomis, M., Klubička, F., Ljubešić, N., Ortiz-Rojas, S., Papavassiliou, V., and Prokopidis, P. (2014). Comparing two acquisition systems for automatically building an english-croatian parallel corpus from multilingual websites. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC'14, Reykjavik, Iceland, may.

Galley, M. and Manning, C. D. (2008). A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 848–856. Association for Computational Linguistics.

Hong, G., Li, C.-H., Zhou, M., and Rim, H.-C. (2010). An empirical study on web mining of parallel data. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING'10, pages 474–

482, Beijing, China. Association for Computational Linguistics.

Jiang, L., Yang, S., Zhou, M., Liu, X., and Zhu, Q. (2009). Mining bilingual data from the web with adaptively learnt patterns. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 2 of *ACL'09*, pages 870–878, Suntec, Singapore.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL*, pages 177–180.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the X Machine Translation Summit*, pages 79–86, Phuket, Thailand.

Ma, X. and Liberman, M. (1999). Bits: A method for bilingual text search over the web. In *Machine Translation Summit VII*, pages 538–542, Singapore, Singapore.

Melamed, D. I. (2001). *Empirical methods for exploiting parallel texts*. MIT Press.

Mollá, D. and Hutchinson, B. (2003). Intrinsic versus extrinsic evaluations of parsing systems. In *Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing: Are Evaluation Methods, Metrics and Resources Reusable?*, pages 43—-50. Association for Computational Linguistics.

Munteanu, D. S. and Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Nie, J.-Y., Simard, M., Isabelle, P., and Durand, R. (1999). Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the Web. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'99, pages 74–81, Berkeley, California, USA. ACM.

Papavassiliou, V., Prokopidis, P., and Thurmair, G. (2013). A modular open-source focused crawler for mining monolingual and bilingual corpora from the web. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 43–51, Sofia, Bulgaria. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL'02, pages 311–318, Philadelphia, Pennsylvania.

Pecina, P., Toral, A., Papavassiliou, V., Prokopidis, P., Tamchyna, A., Way, A., and van Genabith, J. (2014). Domain adaptation of statistical machine translation with domain-focused web crawling. *Language Resources and Evaluation*, pages 1–47.

Resnik, P. and Smith, N. A. (2003). The Web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

Resnik, P. (1998). Parallel strands: a preliminary investigation into mining the web for bilingual text. In *In In Proceedings of the Third Conference of the Association for Machine Translation in the Americas*, AMTA'98, pages 72–82.

Roukos, S., Graff, D., and Melamed, D. (1995). Hansard French/English. Linguistic Data Consortium. Philadelphia, USA.

San Vicente, I. and Manterola, I. (2012). PaCo2: A fully automated tool for gathering parallel corpora from the web. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, LREC'1, Istanbul, Turkey. European Language Resources Association (ELRA).

Schneider, A., Van Der Sluis, I., and Luz, S. (2010). Comparing intrinsic and extrinsic evaluation of MT output in a dialogue system. In *IWSLT*, pages 329–336.

Smith, J. R., Quirk, C., and Toutanova, K. (2010). Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT'10, pages 403–411, Los Angeles, California.

Smith, J. R., Saint-Amand, H., Plamada, M., Koehn, P., Callison-Burch, C., and Lopez, A. (2013). Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1374–1383.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of AMTA*, pages 223–231.

Spiegler, S. (2013). Statistcs of the common crawl corpus 2012. Technical report, Technical report, SwiftKey.

Sridhar, V. K. R., Barbosa, L., and Bangalore, S. (2011). A scalable approach to building a parallel corpus from the web. In *Interspeech*, pages 2113–2116, Florence, Italy.

Suchomel, V., Pomikálek, J., et al. (2012). Efficient web crawling for large text corpora. In *Proceedings of the 7th Web as Corpus Workshop*, WAC7, pages 39–43.

Tiedemann, J. (2009). News from OPUS - a collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, et al., editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation*, Istanbul, Turkey.

Toral, A., Rubino, R., Esplà-Gomis, M., Pirinen, T., Way, A., and Ramírez-Sánchez, G. (2014). Extrinsic Evaluation of Web-Crawlers in Machine Translation: a Case Study on Croatian–English for the Tourism Domain. In *Proceedings of EAMT*, pages 221–224.

Utiyama, M., Kawahara, D., Yasuda, K., and Sumita, E.

(2009). Mining parallel texts from mixed-language web pages. In *Proceedings of the XII Machine Translation Summit*, Ottawa, Ontario, Canada.

Yan, Z., Feng, Y., Hong, Y., and Yao, J. (2009). Parallel sentences mining from the web. *Journal of Computational Information Systems*, 6:1633–1641.

Zhang, Y., Wu, K., Gao, J., and Vines, P. (2006). Automatic acquisition of Chinese–English parallel corpus from the web. In Mounia Lalmas, et al., editors, *Advances in Information Retrieval*, volume 3936 of *Lecture Notes in Computer Science*, pages 420–431. Springer Berlin Heidelberg.

9 Language Resource References

Ljubešić, Nikola and Esplà-Gomis, Miquel and Ortiz Rojas, Sergio and Klubička, Filip and Toral, Antonio. (2016a). *Croatian–English parallel corpus hrenWaC*. Abu-MaTran, 2.0.

Ljubešić, Nikola and Esplà-Gomis, Miquel and Ortiz Rojas, Sergio and Klubička, Filip and Toral, Antonio. (2016b). *Finnish–English parallel corpus fienWaC*. Abu-MaTran, 1.0.

Ljubešić, Nikola and Esplà-Gomis, Miquel and Ortiz Rojas, Sergio and Klubička, Filip and Toral, Antonio. (2016c). *Serbian–English parallel corpus hrenWaC*. Abu-MaTran, 1.0.

Ljubešić, Nikola and Esplà-Gomis, Miquel and Ortiz Rojas, Sergio and Klubička, Filip and Toral, Antonio. (2016d). *Slovene–English parallel corpus hrenWaC*. Abu-MaTran, 1.0.

Nikola Ljubešić. (2009). *SETimes*. Natural Language Processing group, Department of Information and Communication Sciences, University of Zagreb.

M. Cettolo, C. Girardi, M. Federico. (2015). *TED talks*. Web Inventory of Transcribed and Translated Talks.

Steinberger, Ralf and Ebrahim, Mohamed and Poulis, Alexandros and Carrasco-Benitez, Manuel and Schlüter, Patrick and Przybyszewski, Marek and Gilbro, Signe. (2014). *JRC-Aquis*. Joint Research Centre, ISLRN 821-325-977-001-1.

Steinberger, Ralf and Eisele, Andreas and Klocek, Szymon and Pilos, Spyridon and Schlüter, Patrick. (2015). *DGT translation memory*. Joint Research Centre, ISLRN 710-653-952-884-4.

Jörg Tiedemann. (2013). *OpenSubtitles*. OPUS project.